

**SYSTEME DE CONCEPTION ET D'UTILISATION**  
**DE MODELES DECISIONNELS**

La présente invention se rapporte au domaine de  
5 l'intelligence artificielle et plus particulièrement au  
domaine des modèles décisionnels.

La présente invention se rapporte plus  
particulièrement à un système de conception et d'utilisation  
de modèles décisionnels, un modèle décisionnel étant un  
10 modèle mathématique capable de prendre des décisions. Le  
système selon la présente invention possède des applications  
dans le domaine des jeux vidéos, cet exemple restant bien  
entendu non limitatif.

15 L'art antérieur connaît déjà, par la demande de brevet  
français FR 2811449 (Mathématiques Appliquées S.A.), un  
système automatique pour la prise de décision par un agent  
virtuel ou physique. Ce système automatique pour la prise de  
décision par un agent virtuel ou physique en fonction de  
20 variables externes provenant d'un environnement décrit par  
un modèle numérique, et de variables internes à l'agent  
décrites par des paramètres numériques, comprend des moyens  
de sélection d'actions à exercer par l'agent à partir d'une  
variation d'une ou plusieurs desdites variables. Les  
25 paramètres numériques décrivant l'agent virtuel ou physique  
comportent des données numériques représentatives de la  
motivation de l'agent. La sélection d'actions de l'agent  
virtuel ou physique est également fonction de la valeur  
desdites données représentatives de la motivation de  
30 l'agent. Le système présenté dans cette demande de brevet  
PCT présente l'inconvénient de ne traiter uniquement des  
problématiques de choix binaires d'actions. Par exemple, ce  
système n'est pas capable de choisir des paramètres utilisés  
pour la réalisation d'actions. Un autre désavantage de la  
35 solution de Mathématiques Appliquées vis-à-vis de la

présente invention est de ne pas optimiser les ressources informatiques mises à disposition.

La présente invention entend remédier aux inconvénients de l'art antérieur en proposant un système automatique de gestion optimale des ressources logicielles et matérielles utilisées. Le système selon la présente invention permet de plus de déterminer un ensemble idéal de paramètres pour chacune des actions.

10 A cet effet, la présente invention concerne, dans son acception la plus générale, un procédé automatique pour la prise de décision par un agent virtuel ou physique en fonction de variables externes provenant d'un environnement décrit par un modèle numérique ou par des capteurs  
15 physiques, et de variables internes à l'agent décrites par des paramètres numériques, et des moyens de prise de décision commandant les actions dudit agent caractérisé en ce que ledit moyen de prise de décision détermine les paramètres d'une action au moins dudit agent, en fonction  
20 d'une partie au moins des variables internes ou externes, ladite opération de détermination des paramètres d'une action étant réalisée par plusieurs sous-processus de décision fonctions chacun d'une partie au moins desdites variables externes et internes, et traitant une partie  
25 limitée de la problématique décisionnelle générale, les paramètres déterminés par chacun desdits sous-processus étant traités pour construire l'ensemble des paramètres d'action commandant ledit agent.

30 Avantageusement, certaines variables internes sont des valeurs numériques représentatives de la perception.

Selon un mode de mise en œuvre particulier, certaines variables internes sont des valeurs numériques représentatives de la motivation.

Selon une variante, certaines variables externes sont des valeurs numériques représentatives de l'opportunité.

De préférence, le calcul des paramètres, motivations et opportunités est effectué par un arbre de blocs de traitements, chacun des blocs de traitement correspondant à une fonction recevant des variables d'entrée comprenant une partie des variables internes, et des variables externes et des variables de sortie d'un ou de plusieurs blocs de traitements amont.

10       Avantageusement, lesdits blocs de traitements comprennent des opérateurs logiques, des systèmes experts et des opérateurs mathématiques.

De préférence, le procédé comporte un moyen d'interruption de ladite opération de détermination des paramètres consistant à commander l'action de l'agent avec les paramètres déterminés à partir des sous-processus traités avant l'interruption.

15       Avantageusement, l'activation du moyen d'interruption est commandée en fonction de l'état d'un compteur de traitements décrémenté du coût de chaque bloc de traitement utilisé.

De préférence, le coût d'un bloc de traitement est déterminé par un paramètre numérique représentatif du temps-machine nécessaire à l'exécution du traitement dudit bloc.

25       Avantageusement, ledit paramètre numérique représentatif du temps-machine est déterminé de manière relative par rapport aux coûts d'une partie au moins des autres blocs.

Selon un mode de mise en œuvre particulier, ledit moyen d'interruption est commandé par une fonction du système maître.

De préférence, le procédé comporte des étapes d'interrogation du système maître après chaque détermination d'un ensemble de paramètres pour une action, et d'activation

dudit moyen d'interruption en fonction de la réponse du système maître à cette interrogation.

De préférence, le procédé comporte des moyens d'ordonnancement des sous-processus par ordre décroissant de l'activation.

Avantageusement, le procédé comporte des moyens d'enregistrement de l'état des arbres de calcul, des actuateurs et des sous-processus lors de l'interruption, et des moyens pour la poursuite du processus décisionnel à partir des informations enregistrées.

Selon une variante, plusieurs agents sont commandés à partir d'un modèle décisionnel commun, ledit modèle décisionnel comportant un moyen d'enregistrement des informations spécifiques à chacun des agents.

Avantageusement, le procédé ne comporte pas d'étapes d'allocation dynamique de mémoire pendant le traitement du modèle décisionnel.

De préférence, le procédé comporte des moyens de traitement des dépendances logiques entre les sous-processus.

Avantageusement, ledit moyen de traitement des dépendances logiques consiste à modifier l'activation du (ou des) sous-processus maître(s) en fonction de la plus haute activation des sous-processus dépendants.

De préférence, le procédé réalise le traitement de variables externes multivaluées issues de différents objets perçus de l'environnement.

Selon un mode de mise en œuvre particulier, les valeurs de sortie d'un bloc de traitement sont mémorisées au cours du cycle de traitement si elles sont utilisées par un autre sous-processus.

Avantageusement, lesdites valeurs de sorties sont enregistrées dans une mémoire commune à plusieurs blocs de traitement.

De préférence, le procédé comporte un moyen de recalculer une ou plusieurs sorties d'un bloc de traitement dans le cas où la mémoire commune est saturée.

De préférence, le procédé comporte un moyen de  
5 réalisation d'un niveau de détail comportemental.

Avantageusement, le procédé comporte un mécanisme d'adaptation convergent quelle que soit la nature du signal d'apprentissage.

Selon un mode de mise en œuvre particulier,  
10 l'apprentissage est réalisé par un processus stochastique basé sur la distribution de FISK.

De préférence, le procédé comporte des moyens de regroupement et d'enregistrement de souvenirs constitués d'informations correspondant à des perceptions ou à des  
15 résultats d'arbres de calcul.

Avantageusement, le procédé comporte une étape de regroupement des souvenirs sous forme de strates.

Avantageusement, le procédé comporte une étape de reconnaissance consistant à sélectionner les souvenirs  
20 actifs.

Selon une variante avantageuse, le procédé comporte une opération de création d'un nouveau souvenir dans le cas d'absence de reconnaissance parfaite.

Avantageusement, le procédé comporte une opération de  
25 mise à jour des souvenirs actifs par remplacement des informations mémorisées par l'état des informations calculées.

De préférence, le procédé comporte une opération de calcul de la fiabilité des informations des souvenirs mis à  
30 jour.

Selon un mode de mise en œuvre particulier, le procédé comporte des moyens de connexion entre les souvenirs et les actuateurs de la strate.

Avantageusement la connexion s'effectue grâce à des arbres de calcul génériques, communs à tous les souvenirs et utilisant les informations du souvenir.

5 Selon une variante, le procédé comporte des moyens de liaison entre les souvenirs pour la création des influences entre les souvenirs.

Selon une variante, le procédé comporte une étape de propagation des influences entre les souvenirs par le biais de liaisons.

10 Selon un mode de mise en œuvre particulier, le procédé comporte une optimisation des mises à jour des influences limitées aux influences ayant subie une altération.

Selon un mode de mise en œuvre particulier, le procédé comporte une étape de transmission des informations  
15 modifiées au reste du modèle décisionnel.

Avantageusement, le procédé comporte un moyen de transmission des informations modifiées par la création d'un objet virtuel.

De préférence, le procédé comporte un moyen de  
20 transmission des informations modifiées par la surcharge du résultat d'un bloc de traitement.

De préférence, le procédé comporte un mécanisme de gestion automatique du nombre de souvenirs utilisés consistant à supprimer le souvenir le moins utile.

25 Selon une variante avantageuse, ledit mécanisme de gestion automatique du nombre de souvenirs est personnalisable lors de l'étape de conception.

Avantageusement, le procédé comporte un mécanisme de gestion automatique du nombre de liaisons utilisées  
30 consistant à supprimer la liaison la moins utile.

On comprendra mieux l'invention à l'aide de la description, faite ci-après à titre purement explicatif, d'un mode de réalisation de l'invention, en référence aux  
35 figures annexées :

- la figure 1 illustre la factorisation structurée des arbres de calcul ;

- la figure 2 illustre l'utilisation d'un groupe de perceptions ;

5       

- la figure 3 illustre les notions de dépendance logique et de couche ;

- la figure 4 illustre l'ensemble de valeurs lié à un paramètre ;

10       

- la figure 5 illustre la détermination de la valeur optimale d'un paramètre ;

- la figure 6 représente la création par scission d'un nouveau segment d'apprentissage ;

- la figure 7 représente la détermination des deux segments d'apprentissage à fusionner ;

15       

- la figure 8 illustre le mécanisme d'influence entre deux souvenirs par le biais d'une liaison ;

- la figure 9 illustre le mécanisme de création d'un objet virtuel par une liaison.

20       L'architecture du système selon l'invention s'appuie sur la notion de formalisme structurel et fonctionnel. Le modèle est décrit par sa structure et par un nombre déterminé de symboles fonctionnels.

25       Nous allons dans un premier temps donner quelques définitions de symboles fonctionnels afin de faciliter la compréhension de la description d'un mode de réalisation de l'invention.

Nous nommerons dans la suite :

30       

- **centre d'intérêt** un sous-processus ne traitant qu'un seul des aspects du problème décisionnel ;

- **ordre** un résultat issu d'un centre d'intérêt comportant une valeur décrivant son intérêt pour le processus décisionnel, de 0 à n paramètres décrivant la

façon d'exécuter l'ordre, et éventuellement une référence à un objet cible de l'ordre.

5 Dans la suite de la description, les **actions** ou **actuateurs** décrivent les solutions du modèle décisionnel et effectuent les traitements de la détermination de l'ordre optimal.

10 Les **perceptions** ou **senseurs** permettent de récupérer les informations provenant du système maître, indispensables aux traitements des centres d'intérêt.

On entend par **bloc de traitement** ou **module spécifique de traitement** un bloc effectuant une fonction mathématique.

15 A chaque centre d'intérêt est liée la notion de **motivation**. Celle-ci mesure l'intérêt du modèle décisionnel à satisfaire le centre d'intérêt correspondant.

Pour chacun des actuateurs auxquels un centre d'intérêt est lié, on définit l'**opportunité** comme la mesure de l'intérêt d'un élément externe (appartenant à l'environnement) pour le centre d'intérêt correspondant.

20 Pour chacun des actuateurs auxquels un centre d'intérêt est lié, on définit un nombre fini de **paramètres** représentant la configuration à utiliser pour l'exécution de l'actuateur pour le centre d'intérêt correspondant. Un paramètre peut également être qualifié d'**intensité**.

25 L'intérêt de l'ordre est le produit de la motivation du centre d'intérêt par l'opportunité. Il peut donc être différent pour chaque couple (centre d'intérêt, actuateur).

30 Toutes les valeurs (motivation, opportunité, paramètre) sont calculées par des arbres de calcul. Ces arbres de calcul sont des graphes de type arbre dont la racine désigne la valeur à calculer et les feuilles les valeurs utilisées (perceptions) pour ce calcul.

Ces arbres de calcul peuvent effectuer des fonctions mathématiques sur les informations en entrée (perceptions)  
35 au moyen des blocs de traitement. La structure de l'arbre



détermine ainsi l'ensemble des calculs à effectuer. De par la nature graphique de la conception, les calculs sont factorisés en réutilisant un bloc de traitement dans plusieurs arbres de calcul. Ceci permet de conserver la structure minimale (totalement factorisée) des calculs. Ceci est illustré Figure 1.

Dans notre exemple, les perceptions peuvent être représentées par une liste d'objets, chaque objet pouvant être décrit par un ensemble fini de perceptions. Les objets décrits par les mêmes perceptions appartiennent à la même classe d'objets. L'ensemble des perceptions décrivant une classe d'objets forme un **groupe de perceptions**, comme l'illustre la Figure 2.

Pour des raisons d'économies de ressources évidentes, les modèles décisionnels ne peuvent traiter directement des combinaisons d'objets. En effet, si  $n$  objets sont perçus,  $n^2$  combinaisons devraient être traitées. Cependant, il est possible de traiter certaines de ces combinaisons en associant l'un des deux objets de la combinaison à un centre d'intérêt, qui sera alors chargé d'évaluer les combinaisons possibles avec cet objet.

On dit qu'il y a une **dépendance logique** entre deux centres d'intérêt (A dépendant de B) lorsque :

- B est connecté à un actuateur ACT
- ACT modifie le traitement effectué par un arbre de calcul utilisé par A.

Les dépendances logiques cycliques sont automatiquement coupées par le système, la prise en compte de la dépendance n'aura lieu qu'au cycle suivant. La coupure est effectuée une seule fois, au chargement du modèle décisionnel.

Un ensemble d'actuateurs et un ensemble de centres d'intérêt forment une **couche** s'ils sont isolés des autres centres d'intérêt et actuateurs, ou si leurs seuls liens sont des dépendances logiques.

5        La Figure 3 illustre les notions de couche et de dépendance logique.

10        Lorsqu'une couche A voit son traitement modifié par une autre couche B (dépendance logique) au travers d'un actuateur ACT, on dit que B est maître de A. On considère que B envoie des ordres à A. Pour des raisons d'économies de ressources, il est possible de forcer l'actuateur à appeler une fonction du système maître, cette opération remplaçant alors la modification de la couche A. On dit alors que A est  
15 plus détaillée que B. On crée ainsi deux niveaux de détail comportementaux. Un mécanisme adapté mis en place au sein de l'actuateur permet lors de l'utilisation du modèle de spécifier le niveau de détail comportemental désiré.

20        On appelle **cycle de « réflexion »** l'utilisation du modèle décisionnel pour obtenir une décision. Durant le cycle de réflexion, les informations fournies au modèle décisionnel sont invariantes.

25        L'**activation** représente l'importance pour le modèle décisionnel de traiter le centre d'intérêt correspondant. Elle est déterminée en intégrant la motivation du centre d'intérêt par rapport au temps (les cycles de réflexion). Elle est remise à zéro lorsque le centre d'intérêt est traité. Elle favorise l'exécution de tous les centres  
30 d'intérêt selon l'historique de leur motivation.

      Certains centres d'intérêt sont des automates, ils ne possèdent pas de motivation (pas d'activation). Ils héritent de la plus forte activation des centres d'intérêt dépendants. A cet héritage est ajouté un « epsilon » (nombre  
35 positif très proche de zéro). Ce mécanisme garantit que les

dépendances automatiques (centres d'intérêts automates maîtres) sont traitées avant les centres d'intérêt dépendants.

5 La motivation n'est pas connectée à un groupe de perceptions, car elle se rapporte à l'état interne de l'entité considérée utilisant le modèle décisionnel appartenant au système maître. Ainsi, la motivation puis l'activation peuvent être rapidement calculées.

10 Le modèle décisionnel met en œuvre une étape de tri des centres d'intérêt selon leur activation. Les centres d'intérêt sont alors traités dans cet ordre, de la plus forte activation à la plus faible. Ce système permet d'obtenir des décisions cohérentes même si le processus décisionnel est interrompu avant d'avoir traité tous les  
15 centres d'intérêt. Ainsi, les traitements réalisés au sein du système affichent un caractère interruptible tout en maintenant une certaine cohérence.

20 Les traitements réalisés au sein du système selon l'invention sont principalement mis en œuvre par les centres d'intérêts. L'objectif principal des centres d'intérêt est de calculer les ordres transmis aux actuateurs auxquels ils sont connectés. Pour ce faire, ils utilisent les arbres de calcul définis lors de l'étape de conception du modèle  
25 décisionnel.

Les autres traitements sont réalisés au sein des actuateurs. Ils concernent la gestion des ordres reçus.

30 Chaque actuateur formalise une partie du résultat souhaité. Les actuateurs étant indépendants entre eux, un mécanisme local suffit pour la détermination d'un résultat satisfaisant l'ensemble des critères décisionnels. Une étape de concertation entre les actuateurs n'est donc pas nécessaire, ce qui conduit naturellement à une optimisation  
35 des calculs mis en œuvre.

Un mécanisme local d'influence entre les ordres permet à l'actuateur de déterminer une solution globale. Pour mesurer l'influence d'un ordre A sur un ordre B, l'actuateur calcule la complémentarité de A vis-à-vis de B. Cette  
5 complémentarité est comprise entre 0 (ordres incompatibles) et 1 (ordres totalement complémentaires). Elle pondère l'intérêt de A reçu par B. La complémentarité est calculée en faisant appel à une fonction définie dans le système maître.

10 L'ordre qui possède après influence l'intérêt le plus élevé sera exécuté. Avant cela, il est nécessaire de calculer les valeurs optimales de ses paramètres. Chaque paramètre est décrit par trois valeurs :

- la valeur initiale  $p$  du paramètre
- 15 - une valeur  $m$  représentant une marge dans laquelle les valeurs du paramètre sont considérées comme idéales
- une valeur  $s$  représentant une marge de sécurité dans laquelle les valeurs du paramètre sont éligibles mais moins intéressantes.

20 La Figure 4 représente l'ensemble des valeurs possibles décrit par  $(p, m, s)$ .

La valeur optimale de chaque paramètre est comprise entre  $(p-m-s)$  et  $(p+m+s)$ . On utilise l'intérêt comme critère de sélection. On calcule, pour chaque valeur possible,  
25 l'intérêt cumulé (provenant des différents ordres). La valeur optimale est alors désignée par la valeur possédant l'intérêt cumulé le plus important. S'il s'agit d'un segment, soit l'une des deux valeurs extrêmes, soit la valeur correspondant au centre du segment est sélectionnée.

30 La Figure 5 illustre le calcul de la valeur optimale d'un paramètre.

Pour des raisons de coût mémoire, chaque actuateur ne peut gérer qu'un nombre limité d'ordres (nombre fixé par  
35 l'utilisateur lors de l'étape de conception). Un procédé

interne de fusion des ordres permet de respecter cette limite sans perdre trop d'informations. Ce procédé supprime l'ordre le moins intéressant et possédant la complémentarité la plus élevée. Son influence sur l'ordre dont il est complémentaire est définitivement intégrée dans ce dernier. En d'autres termes, il s'agit de déterminer le couple d'ordres (A,B) pour lequel la complémentarité est la plus élevée. On supprime alors l'ordre A et on modifie l'ordre B afin de prendre en compte l'intérêt de A. Si deux couples (A, B) et (C,D) possèdent les mêmes complémentarités, on supprime de A ou de C celui qui possède le plus faible intérêt.

Chaque actuateur intègre une notion de persistance. Il existe trois persistances possibles, qui modifient son fonctionnement :

- **volatile** : les ordres sont exécutés directement dès qu'ils sont reçus par l'actuateur sans qu'aucun mécanisme d'influence n'intervienne. Ils sont ensuite effacés du système.

- **Persistent** : les ordres ne sont conservés que durant le cycle de réflexion. Ils sont effacés au début de chaque nouveau cycle.

- **Statique** : les ordres sont conservés entre les cycles de réflexion. A chaque ordre est associé un arbre de calcul particulier nommé **satiété** (comprise entre 0 et 1). Au début de chaque nouveau cycle, le complémentaire à 1 de la valeur de la satiété pondère l'intérêt de l'ordre. Si la satiété est élevée, l'intérêt est faible, ce qui provoque la fusion de l'ordre. Un ordre d'intérêt nul est immédiatement effacé.

Après avoir décrit le mécanisme de sélection d'actions, nous allons nous attacher dans une seconde partie à examiner le mécanisme d'adaptation.

Le mécanisme d'adaptation mis en place est localisé et contrôlé :

- **Localisation** : Le mécanisme d'adaptation est mis en œuvre au niveau de certains blocs de traitement particuliers appelés « blocs de traitement adaptatifs ». Ainsi, seuls les arbres de calculs utilisant ces blocs seront susceptibles de s'adapter.
- **Contrôle** : Le mécanisme d'adaptation est déclenché par un actuateur particulier appelé « surveillant ». Ses paramètres représentent chacun un jugement porté sur la situation actuelle.

Le jugement ou signal d'apprentissage est représenté par une valeur pouvant être de trois natures différentes :

- **Echec** : plus la valeur est élevée, plus la situation jugée est défavorable.
- **Réussite** : plus la valeur est élevée, plus la situation jugée est favorable.
- **Echec dichotomique** : en plus de mesurer un échec, le jugement fournit une indication sur le sens de l'adaptation. Cette nature ne peut être utilisée que pour les arbres de calcul liés aux paramètres (à valeur exacte et non à valeur relative, cette dernière décrivant la motivation et l'opportunité).

La nature mathématique du jugement (continu ou discret, simple ou dichotomique) influence de manière significative la convergence de l'adaptation. Le mécanisme d'adaptation mis en place au niveau du bloc de traitement adaptatif doit absolument fonctionner quelle que soit la nature du jugement reçu.

L'adaptation au sein des blocs de traitement adaptatif s'effectue en faisant varier les paramètres d'apprentissage. Ces paramètres sont utilisés lors du traitement effectué par

le bloc. En les faisant varier, on modifie ce traitement. Les valeurs possibles de ces paramètres doivent appartenir à un ensemble fini, borné et ordonné.

Chaque bloc de traitement adaptatif est caractérisé par une **fiabilité** qui mesure l'adéquation du bloc avec les situations rencontrées. Elle est comprise entre 0 et 1. Elle est calculée selon la formule suivante :

Soit  $e$  l'erreur mesurée

$$Fiabilité_{n+1} = \frac{Fiabilité_n * n + 1 - e}{n + 1}$$

10

Si  $ValAbs(Fiabilité_{n+1} - Fiabilité_n) > \epsilon$ , alors  $n = n + 1$ .

La nouvelle valeur d'un paramètre d'apprentissage est calculée selon la formule :

15

Soit  $e$  l'erreur associée au signal,

Soit  $f$  la fiabilité du bloc,

Soit  $a$  la valeur actuelle du paramètre,

Soit  $p$  un nombre aléatoire compris entre 0 et 1,

$$Valeur = a + e * (1 - f) * \sqrt{\frac{p}{1 - p}}$$

20

La nouvelle valeur ainsi calculée suit la distribution dite de Fisk, qui peut être rendue symétrique.

Considérons trois exemples de blocs d'apprentissages :

25

- Le bloc d'apprentissage pondération

- Le bloc d'apprentissage pluri linéaire

- Le bloc d'apprentissage neural.

Le bloc d'apprentissage pondération ne possède qu'un seul paramètre d'apprentissage. Ce bloc possède deux modes de fonctionnement selon qu'il possède ou non une entrée :

30

- Mode pondération : si le bloc possède une entrée, il pondère alors son entrée par son paramètre d'apprentissage (produit)

- Mode génération : si le bloc ne possède pas d'entrée, il renvoie simplement son paramètre d'apprentissage, comme si son entrée était 1.

L'apprentissage du paramètre d'apprentissage s'effectue simplement en appliquant à la lettre les indications précédentes. Lorsque le bloc reçoit un signal d'apprentissage :

- Il utilise la moyenne des valeurs d'entrée références

- La référence d'apprentissage est obtenue en multipliant cette moyenne par le paramètre d'apprentissage actuel

- L'échelle s'obtient classiquement en fonction de la fiabilité et du signal

- Le nouveau paramètre d'apprentissage est obtenu en divisant la valeur apprise par la moyenne des valeurs d'entrée références.

Soit  $i$  la moyenne des valeurs d'entrée références,  
Soit  $w$  la valeur du paramètre d'apprentissage,  
Soit  $e$  l'erreur associée au signal,  
Soit  $f$  la fiabilité du bloc d'apprentissage,  
Soit  $w'$  la nouvelle valeur du paramètre d'apprentissage,  
Soit  $a$  la localisation,  
Soit  $b$  l'échelle,

On a :  $a = w*i$

$$b = e*(1-f)$$
$$w' = (a + b*fisk) / i$$

et ceci, quel que soit  $i$  non nul.



Concernant le bloc d'apprentissage pluri linéaire, le concept est de décomposer la fonction idéale en un ensemble de segments, appelés segments d'apprentissage. Chaque segment d'apprentissage approxime une portion continue de la fonction idéale. Il possède ses propres paramètres d'apprentissage : pente et ordonnée à l'origine. Ce bloc d'apprentissage regroupe ainsi plusieurs paramètres d'apprentissage : les pentes et les ordonnées à l'origine des segments d'apprentissage.

Soit  $a$  la pente,

Soit  $b$  l'ordonnée à l'origine,

$$y = a * x + b$$

Chaque segment d'apprentissage comprend deux limites de définition. Il enregistre une moyenne des valeurs d'entrée utilisées correspondant à ses limites de définition.

Le segment calcule la valeur de sortie idéale comme précédemment. La valeur d'entrée mémorisée correspond en fait à la moyenne des valeurs d'entrée utilisées. Un nouveau point d'apprentissage est créé. Le bloc d'apprentissage effectue alors une scission du segment d'apprentissage concerné au point d'entrée utilisé. Il crée ainsi deux segments d'apprentissage à partir d'un seul. La figure 6 résume le fonctionnement de cette scission.

Les paramètres d'apprentissage sont alors calculés comme suit :

Soient  $x_1$  et  $y_1$  les coordonnées du premier point d'apprentissage,

Soient  $x_2$  et  $y_2$  les coordonnées du second point d'apprentissage,

Soit  $a$  la pente de l'équation linéaire,

Soit  $b$  l'ordonnée à l'origine de l'équation linéaire,

5  $a = (y_2 - y_1)/(x_2 - x_1)$  si deux points d'apprentissage

$a = 0$  sinon

$b = x_1*a + y_1$  si un point d'apprentissage

$b = 0$  sinon

10 Le nombre maximum de segments est défini lors de l'étape de conception.

Lorsque ce nombre est dépassé, le bloc d'apprentissage procède à une « fusion ». La fusion consiste à choisir l'un des points délimitant deux segments d'apprentissage et à  
15 l'éliminer. Les deux segments adjacents recalculent alors leur paramètre d'apprentissage en conséquence. La fusion correspond à une perte d'information pour le bloc. Il va donc minimiser cette perte. Cette perte est liée à plusieurs facteurs :

20 • Fiabilité du segment d'apprentissage : plus le segment est fiable, moins il est préférable de le modifier.

• Surface du triangle formé par les extrémités du nouveau segment et le point à éliminer : cette surface représente l'information perdue lors de l'élimination perdue  
25 du point concerné.

Elle se calcule selon la formule :

Soit  $f$  la fiabilité,

30 Soit  $l$  la différence entre les abscisses,

Soit  $h$  la différence entre les ordonnées,

$$\text{Information} = (1/2)*f*l*h$$

La Figure 7 montre comment s'applique ce processus de fusion.

Le bloc d'apprentissage neural se décompose en deux fonctions. La première utilise les paramètres d'apprentissage pour pondérer les entrées du bloc : un paramètre d'apprentissage par entrée. Elle les combine en effectuant une somme. La seconde permet de seuiller la sortie. Par défaut, cette sortie est une sigmoïde.

10

Soit  $f_{mod}$  la fonction de modification du résultat,  
Soit  $f_{comb}$  la fonction de combinaison des entrées,  
Soit  $k$  la raideur de la sigmoïde

15

$$f_{mod}(x) = (\exp(k \cdot x) - 1) / (\exp(k \cdot x) + 1)$$

$$f_{comb}(x) = \sum_i \text{entrée}_i * w_i$$

L'apprentissage s'effectue pour chacune des entrées. Le signal d'apprentissage reçu par le bloc d'apprentissage est donc utilisé par toutes les entrées. La référence d'apprentissage est la moyenne des valeurs utilisées depuis le dernier signal d'apprentissage. L'échelle se calcule comme précédemment.

Le résultat de l'apprentissage est alors une nouvelle valeur pour le paramètre d'apprentissage. Deux grandes différences de fonctionnement distinguent le bloc d'apprentissage neural, d'un neurone classique :

- L'utilisation de la fiabilité : les réseaux de neurones utilisent un facteur d'apprentissage similaire, hérité des méthodes de recuit simulé (alors appelé température). La fiabilité permet de remplacer ce paramètre avantageusement. Sa valeur dépend en effet des signaux d'apprentissage reçus, et n'est donc pas arbitraire.

• Un apprentissage stochastique : il est possible de rendre le bloc d'apprentissage neural non stochastique. Cependant, la convergence peut alors se heurter à la pauvreté de certains signaux d'apprentissage. Les réseaux de neurones conventionnels utilisent en effet un signal d'apprentissage continu mesurant l'échec.

Après avoir décrit les mécanismes de sélection d'actions et d'adaptation, nous allons étudier la modélisation et la planification dynamique.

On distingue deux concepts permettant la création d'un modèle aussi riche que souhaité :

• Le souvenir : il permet de mémoriser un ensemble d'informations. Cet ensemble représente un concept du système maître. Chacun de ces ensembles d'informations décrit une classe de souvenirs.

• La liaison : elle permet de décrire les influences qui modifient les souvenirs liés entre eux.

La liaison s'apparente en réalité à un actuateur qui va modifier les informations d'un souvenir. Ces modifications sont engendrées par les souvenirs qui vont donc envoyer des ordres à l'actuateur liaison. Les souvenirs sont donc des centres d'intérêt.

Chaque souvenir appartient à une classe déterminée qui spécifie les informations qu'il contient. Ces classes sont appelées classes de souvenir. Tous les souvenirs appartenant à une même classe forment une strate de souvenirs.

Les souvenirs appartenant à une même strate sont créés par un actuateur appelé « actuateur mémoriser ». Chacun de ces paramètres détermine la valeur d'une information du souvenir.

Les informations mémorisées dans un souvenir proviennent d'arbres de calcul. Elles sont donc plus ou moins directement issues des percepteurs.

« L'actuateur mémoriser », lors de son exécution, propose aux souvenirs existants la création d'un nouveau souvenir. Chaque souvenir existant va alors essayer de s'identifier dans le souvenir proposé : c'est l'étape de reconnaissance.

On appelle **objet référence** du souvenir ou plus simplement référence du souvenir un identifiant provenant du système maître et désignant l'objet source qui est à l'origine du souvenir.

La reconnaissance s'effectue au moyen d'un arbre de calcul spécifique appelé arbre de reconnaissance, conçu pour chaque strate de souvenirs. En cas d'absence de cet arbre, le système procède à une reconnaissance directe par référence.

Si aucun souvenir ne se reconnaît parfaitement dans le souvenir proposé, un nouveau souvenir est créé. Ce nouveau souvenir s'identifie donc parfaitement au souvenir proposé.

Tous les souvenirs parvenant à s'identifier, même partiellement, au souvenir proposé deviennent les **souvenirs actifs**. Ces souvenirs ne peuvent transmettre leur influence aux autres souvenirs. Ce sont les réceptacles des influences des autres souvenirs.

Les souvenirs actifs modifient leurs informations pour correspondre parfaitement au souvenir proposé. Cette étape correspond à la mise à jour des souvenirs.

Chaque information regroupe trois valeurs distinctes :  
• L'information mémorisée (information de référence) : cette valeur correspond au résultat de l'arbre de calcul lors de la dernière mémorisation.

• L'information modifiée (information chargée) : cette valeur correspond à l'information mémorisée (information de référence), modifiée par les souvenirs liés.

• La fiabilité de l'information : cette valeur permet de mesurer la stabilité dans le temps de l'information mémorisée. Elle se calcule selon la formule :

Soit  $e$  l'erreur mesurée

5

$$Fiabilité_{n+1} = \frac{Fiabilité_n * n + 1 - e}{n + 1}$$

Si  $ValAbs(Fiabilité_{n+1} - Fiabilité_n) > \epsilon$ , alors  $n = n + 1$ .

10

La fiabilité globale d'un souvenir se calcule en effectuant la moyenne des fiabilités de ses informations.

Les souvenirs sont automatiquement connectés lors de leur création aux actuateurs présents dans la strate. Ces actuateurs sont appelés « **actuateurs de strate** ».

Les arbres de calcul nécessaires au calcul des ordres peuvent utiliser les informations des souvenirs grâce à des percepteurs spécifiques. Ces arbres de calcul sont dits génériques, puisqu'ils sont partagés par tous les souvenirs.

L'objet source d'un ordre envoyé à un actuateur de strate est toujours la référence du souvenir qui est à l'origine de cet ordre.

Parmi les actuateurs de strate, les « **actuateurs lier** » permettent de créer une liaison vers un souvenir cible. Un seul actuateur lié permet de gérer l'ensemble des souvenirs cibles. Ces actuateurs sont des actuateurs génériques. A chaque création d'un nouveau souvenir, l'actuateur lié correspondant est mis à jour.

Les liaisons permettent à un souvenir d'en influencer un autre, c'est-à-dire de modifier ses informations (l'information est alors dite chargée), comme l'illustre la Figure 8.

Les liaisons dont la cible est un souvenir actif ne propagent pas leur influence sur ce souvenir. Par contre, elles utilisent leur influence pour modifier la couche qui va utiliser les souvenirs. Cette modification peut prendre  
5 deux formes :

- la création d'un objet virtuel : un nouvel objet qui n'existe pas réellement dans le système maître va être créé et perçu par la couche ;

- la surcharge d'un bloc de traitement : le résultat  
10 d'un bloc de traitement utilisé par la couche est remplacé, le système se met alors à jour automatiquement.

La Figure 9 représente la création d'un objet virtuel par une liaison.

15

Le calcul de l'ensemble des influences est un calcul long, d'une complexité de l'ordre de  $n^2$ ,  $n$  étant le nombre de souvenirs.

Cependant, le système est capable de ne gérer que les  
20 mises à jour nécessaires, correspondant à un changement de souvenirs actifs ou à une modification des informations d'un souvenir. Ce procédé garantit donc un traitement incrémental qui peut être réparti dans le temps.

25 Les strates de souvenirs sont capables de gérer automatiquement les ressources mémoire qu'elles consomment. La création des souvenirs et des liaisons ayant déjà été abordée, un actuateur de strate spécifique appelé « **actuateur oublier** » permet de gérer leur destruction.

30 Il est important de noter que cet actuateur existe, qu'il ait été créé ou non lors de l'étape de conception. Son comportement par défaut consiste à éliminer le souvenir et/ou la liaison la moins utile.

L'utilité d'un souvenir dépend de deux facteurs  
35 principaux :

- l'opportunité calculée lors de la création ou dernière mise à jour du souvenir permet de connaître son utilité pour le modèle décisionnel.

- le nombre de mises à jour du souvenir pondéré par sa  
5 fiabilité permet de connaître la « certitude » du souvenir.

L'utilité d'un souvenir se calcule donc selon la formule :

Utilité = (Opportunité) \* (Fiabilité) \* (Nombre de mises à jours)

10 L'utilité d'une liaison dépend de deux facteurs :

- utilité du souvenir cible

- opportunité calculée pour la création de la liaison, qui mesure l'intérêt de cette liaison pour le modèle décisionnel.

15 L'utilité d'une liaison se calcule donc selon la formule :

Utilité = (Opportunité) \* (Utilité(Souvenir cible))

20 Il est possible lors de l'étape de conception de spécifier un « actuateur oublier ». Cet actuateur possède un seul paramètre qui détermine un pourcentage de la fiabilité du souvenir à conserver.

25 L'invention est décrite dans ce qui précède à titre d'exemple. Il est entendu que l'homme du métier est à même de réaliser différentes variantes de l'invention sans pour autant sortir du cadre du brevet.



**REVENDICATIONS**

1. Procédé automatique pour la prise de décision par un agent virtuel ou physique en fonction de variables  
5 externes provenant d'un environnement décrit par un modèle numérique ou par des capteurs physiques, et de variables internes à l'agent décrites par des paramètres numériques, et des moyens de prise de décision commandant les actions dudit agent caractérisé en ce que ledit moyen de prise de  
10 décision détermine les paramètres d'une action au moins dudit agent, en fonction d'une partie au moins des variables internes ou externes, ladite opération de détermination des paramètres d'une action étant réalisée par plusieurs sous-processus de décision fonctions chacun d'une partie au moins  
15 desdites variables externes et internes, et traitant une partie limitée de la problématique décisionnelle générale, les paramètres déterminés par chacun desdits sous-processus étant traités pour construire l'ensemble des paramètres d'action commandant ledit agent.

20

2 - Procédé automatique selon la revendication 1, caractérisé en ce que certaines variables internes sont des valeurs numériques représentatives de la perception.

25

3 - Procédé automatique selon la revendication 1, caractérisé en ce que certaines variables internes sont des valeurs numériques représentatives de la motivation.

30 4 - Procédé automatique selon la revendication 1, caractérisé en ce que certaines variables externes sont des valeurs numériques représentatives de l'opportunité.

5 5 - Procédé automatique selon la revendication 1, caractérisé en ce que le calcul des paramètres, motivations

et opportunités est effectué par un arbre de blocs de traitements, chacun des blocs de traitement correspondant à une fonction recevant des variables d'entrée comprenant une partie des variables internes, et des variables externes et  
5 des variables de sortie d'un ou de plusieurs blocs de traitements amont.

6 - Procédé automatique selon la revendication 5, caractérisé en ce que lesdits blocs de traitements  
10 comprennent des opérateurs logiques, des systèmes experts et des opérateurs mathématiques.

7 - Procédé automatique selon la revendication 1, caractérisé en ce qu'il comporte un moyen d'interruption de  
15 ladite opération de détermination des paramètres consistant à commander l'action de l'agent avec les paramètres déterminés à partir des sous-processus traités avant l'interruption.

20 8 - Procédé automatique selon la revendication 7, caractérisé en ce que l'activation du moyen d'interruption est commandée en fonction de l'état d'un compteur de traitements décrémenté du coût de chaque bloc de traitement utilisé.

25

9 - Procédé automatique selon la revendication 8, caractérisé en ce que le coût d'un bloc de traitement est déterminé par un paramètre numérique représentatif du temps-machine nécessaire à l'exécution du traitement dudit bloc.

30 10 - Procédé automatique selon la revendication 9, caractérisé en ce que ledit paramètre numérique représentatif du temps-machine est déterminé de manière relative par rapport aux coûts d'une partie au moins des autres blocs.

35

11 - Procédé automatique selon la revendication 7, caractérisé en ce que ledit moyen d'interruption est commandé par une fonction du système maître.

5 12 - Procédé automatique selon la revendication 7, caractérisé en ce qu'il comporte des étapes d'interrogation du système maître après chaque détermination d'un ensemble de paramètres pour une action, et d'activation dudit moyen d'interruption en fonction de la réponse du système maître à  
10 cette interrogation.

13 - Procédé automatique selon la revendication 7, caractérisé en ce qu'il comporte des moyens d'ordonnancement des sous-processus par ordre décroissant de l'activation.

15 14 - Procédé automatique selon la revendication 7, caractérisé en ce qu'il comporte des moyens d'enregistrement de l'état des arbres de calcul, des actuateurs et des sous-processus lors de l'interruption, et des moyens pour la  
20 poursuite du processus décisionnel à partir des informations enregistrées.

15 - Procédé automatique selon la revendication 1, caractérisé en ce que plusieurs agents sont commandés à  
25 partir d'un modèle décisionnel commun, ledit modèle décisionnel comportant un moyen d'enregistrement des informations spécifiques à chacun des agents.

16 - Procédé automatique selon la revendication 1,  
30 caractérisé en ce qu'il ne comporte pas d'étapes d'allocation dynamique de mémoire pendant le traitement du modèle décisionnel.

17 - Procédé automatique selon la revendication 1, caractérisé en ce qu'il comporte des moyens de traitement des dépendances logiques entre les sous-processus.

5        18 - Procédé automatique selon la revendication 17, caractérisé en ce que ledit moyen de traitement des dépendances logiques consiste à modifier l'activation du (ou des) sous-processus maître(s) en fonction de la plus haute activation des sous-processus dépendants.

10

19 - Procédé automatique selon la revendication 1, caractérisé en ce qu'il réalise le traitement de variables externes multivaluées issues de différents objets perçus de l'environnement.

15

20 - Procédé automatique selon la revendication 5, caractérisé en ce que les valeurs de sortie d'un bloc de traitement sont mémorisées au cours du cycle de traitement si elles sont utilisées par un autre sous-processus.

20

21 - Procédé automatique selon la revendication 20, caractérisé en ce que lesdites valeurs de sorties sont enregistrées dans une mémoire commune à plusieurs blocs de traitement.

25

22 - Procédé automatique selon la revendication 21, caractérisé en ce qu'il comporte un moyen de recalculer une ou plusieurs sorties d'un bloc de traitement dans le cas où la mémoire commune est saturée.

30

23 - Procédé automatique selon la revendication 1, caractérisé en ce qu'il comporte un moyen de réalisation d'un niveau de détail comportemental.

24 — Procédé automatique selon la revendication 1, caractérisé en ce qu'il comporte un mécanisme d'adaptation convergent quelle que soit la nature du signal d'apprentissage.

5

25 — Procédé automatique selon la revendication 24, caractérisé en ce que l'apprentissage est réalisé par un processus stochastique basé sur la distribution de FISK.

10

26 — Procédé automatique selon la revendication 1, caractérisé en ce qu'il comporte des moyens de regroupement et d'enregistrement de souvenirs constitués d'informations correspondant à des perceptions ou à des résultats d'arbres de calcul.

15

27 — Procédé automatique selon la revendication 26, caractérisé en ce qu'il comporte une étape de regroupement des souvenirs sous forme de strates.

20

28 — Procédé automatique selon la revendication 26, caractérisé en ce qu'il comporte une étape de reconnaissance consistant à sélectionner les souvenirs actifs.

25

29 — Procédé automatique selon la revendication 28, caractérisé en ce qu'il comporte une opération de création d'un nouveau souvenir dans le cas d'absence de reconnaissance parfaite.

30

30 — Procédé automatique selon la revendication 28, caractérisé en ce qu'il comporte une opération de mise à jour des souvenirs actifs par remplacement des informations mémorisées par l'état des informations calculées.

31 — Procédé automatique selon la revendication 30, caractérisé en ce qu'il comporte une opération de calcul de la fiabilité des informations des souvenirs mis à jour.

5        32 — Procédé automatique selon la revendication 27, caractérisé en ce qu'il comporte des moyens de connexion entre les souvenirs et les actuateurs de la strate.

10       33 — Procédé automatique selon la revendication 32, caractérisé en ce que la connexion s'effectue grâce à des arbres de calcul génériques, communs à tous les souvenirs et utilisant les informations du souvenir.

15       34 — Procédé automatique selon la revendication 26, caractérisé en ce qu'il comporte des moyens de liaison entre les souvenirs pour la création des influences entre les souvenirs.

20       35 — Procédé automatique selon la revendication 34, caractérisé en ce qu'il comporte une étape de propagation des influences entre les souvenirs par le biais de liaisons.

25       36 — Procédé automatique selon la revendication 35, caractérisé en ce qu'il comporte une optimisation des mises à jour des influences limitées aux influences ayant subie une altération.

30       37 — Procédé automatique selon la revendication 35, caractérisé en ce qu'il comporte une étape de transmission des informations modifiées au reste du modèle décisionnel.

35       38 — Procédé automatique selon la revendication 37, caractérisé en ce qu'il comporte un moyen de transmission des informations modifiées par la création d'un objet virtuel.

39 - Procédé automatique selon la revendication 37,  
caractérisé en ce qu'il comporte un moyen de transmission  
des informations modifiées par la surcharge du résultat d'un  
5 bloc de traitement.

40 - Procédé automatique selon la revendication 26,  
caractérisé en ce qu'il comporte un mécanisme de gestion  
automatique du nombre de souvenirs utilisés consistant à  
10 supprimer le souvenir le moins utile.

41 - Procédé automatique selon la revendication 40,  
caractérisé en ce que ledit mécanisme de gestion automatique  
du nombre de souvenirs est personnalisable lors de l'étape  
15 de conception.

42 - Procédé automatique selon la revendication 35,  
caractérisé en ce qu'il comporte un mécanisme de gestion  
automatique du nombre de liaisons utilisées consistant à  
20 supprimer la liaison la moins utile.

1/5

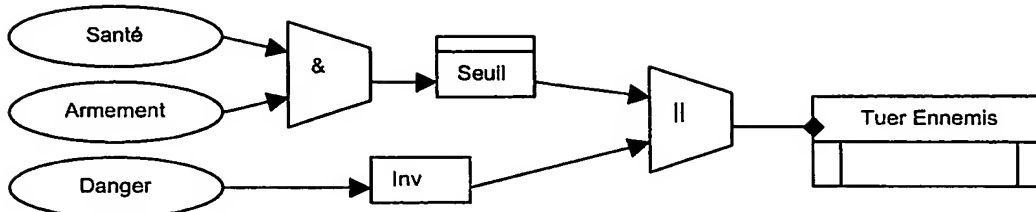


Figure 1

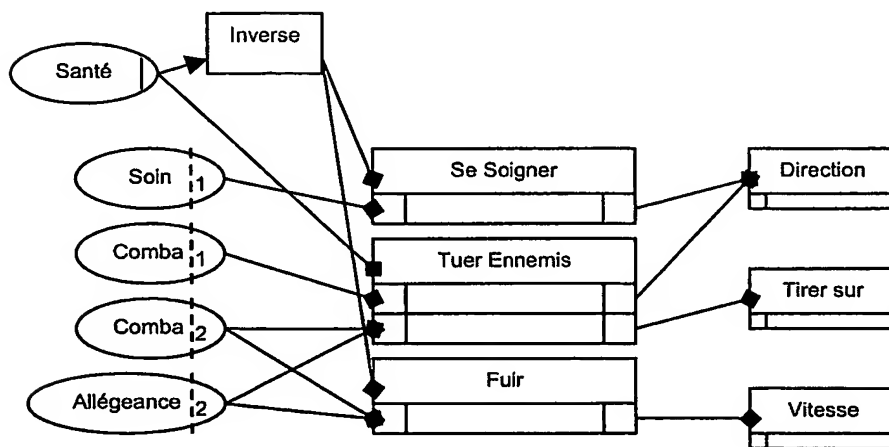


Figure 2

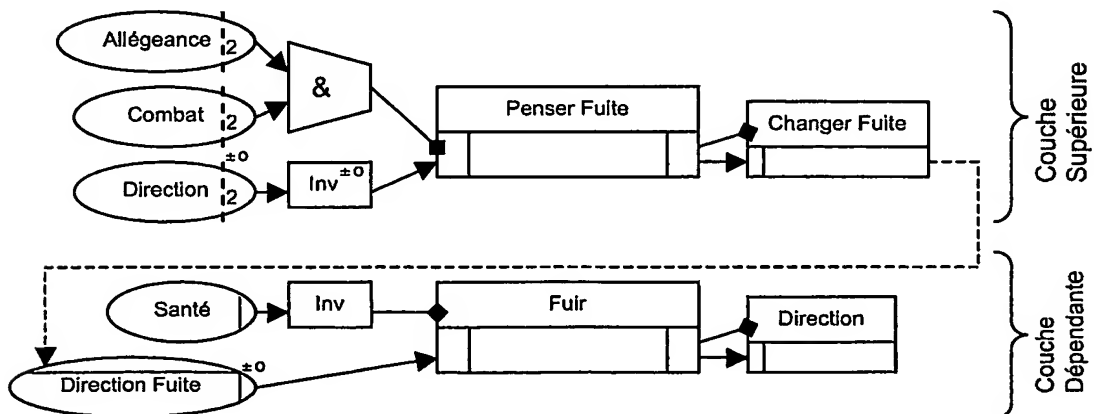


Figure 3



2/5

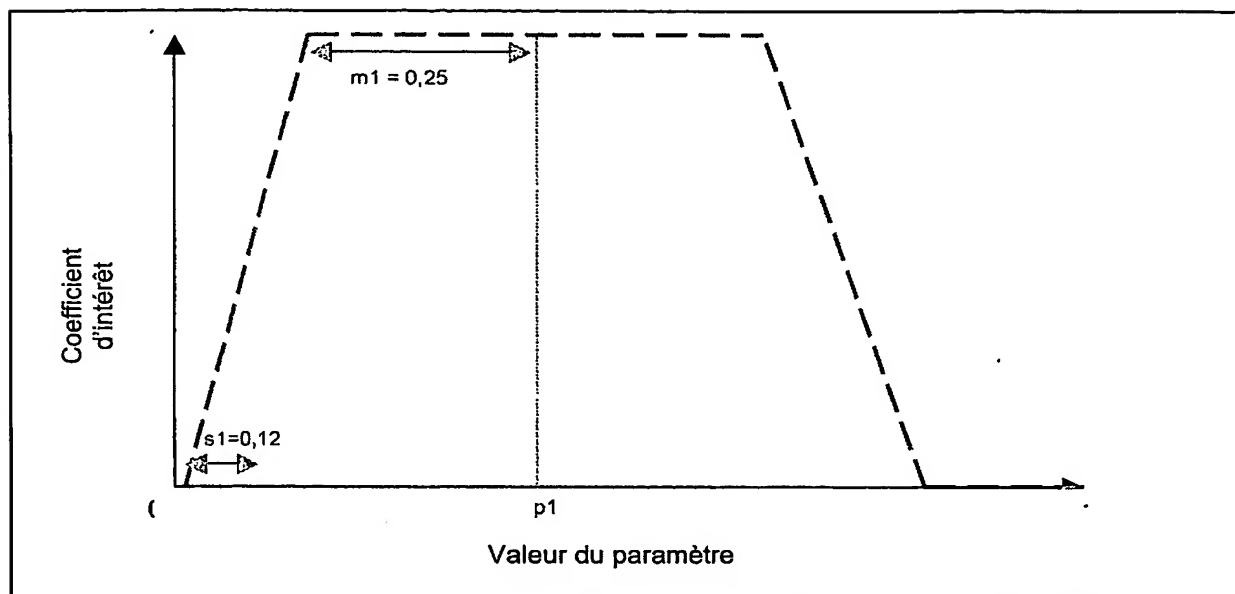


Figure 4

3/5

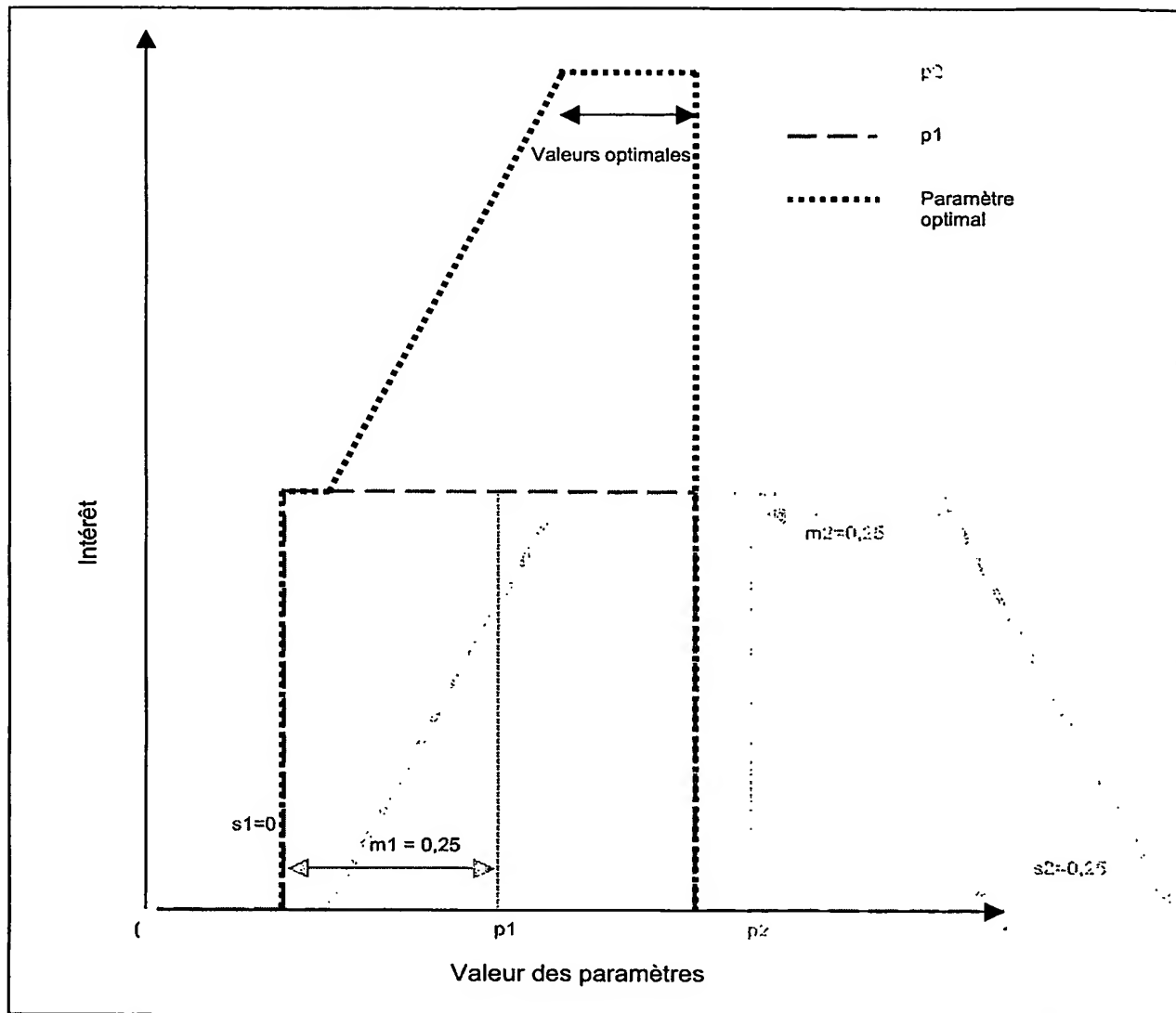


Figure 5

Figure 6

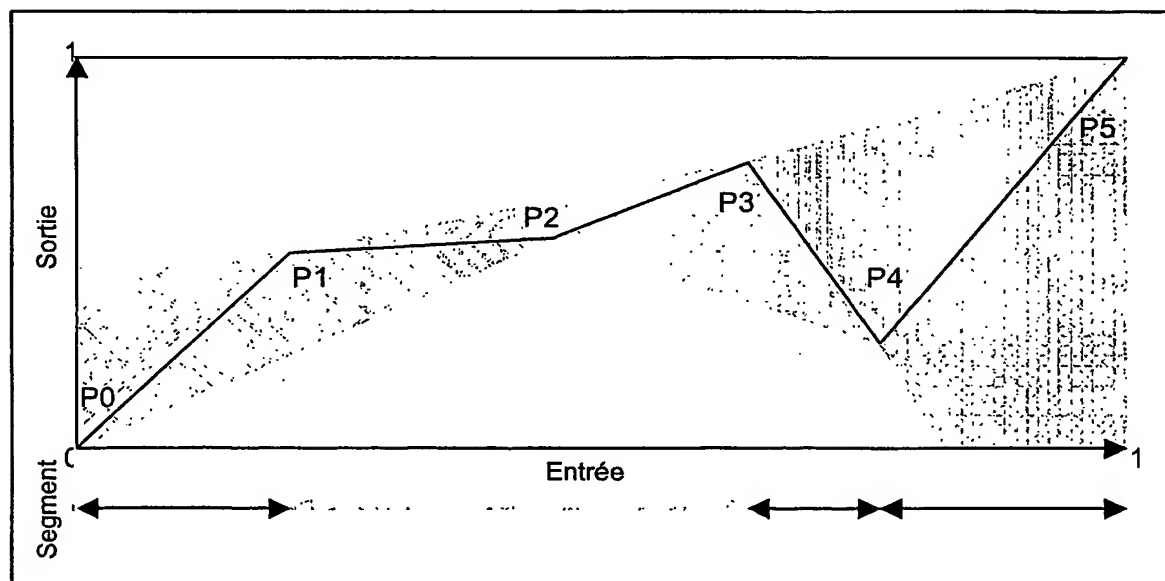
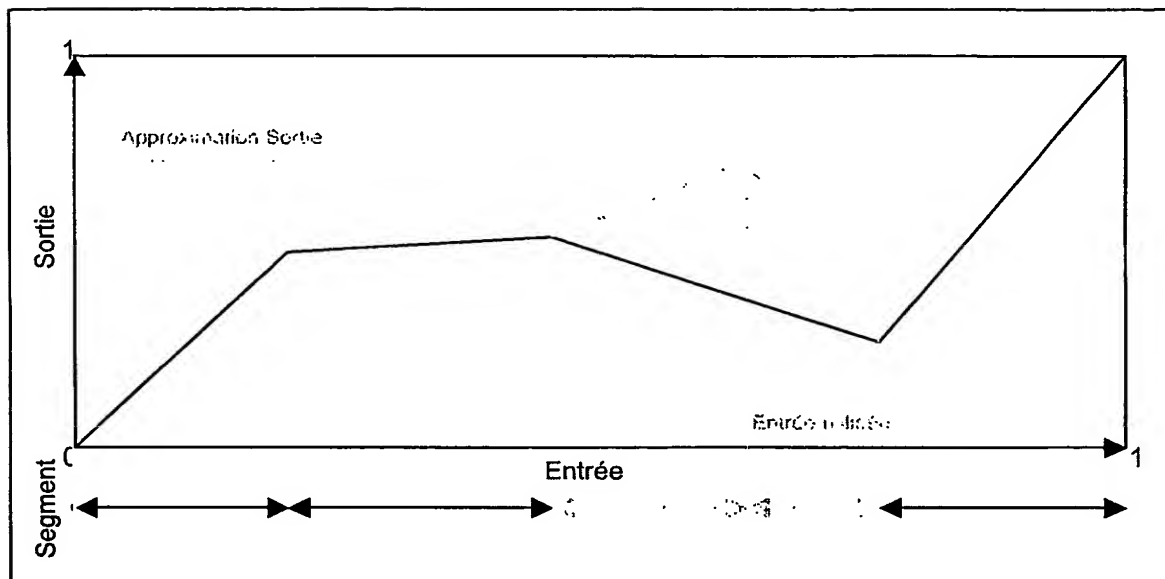


Figure 7

5/5

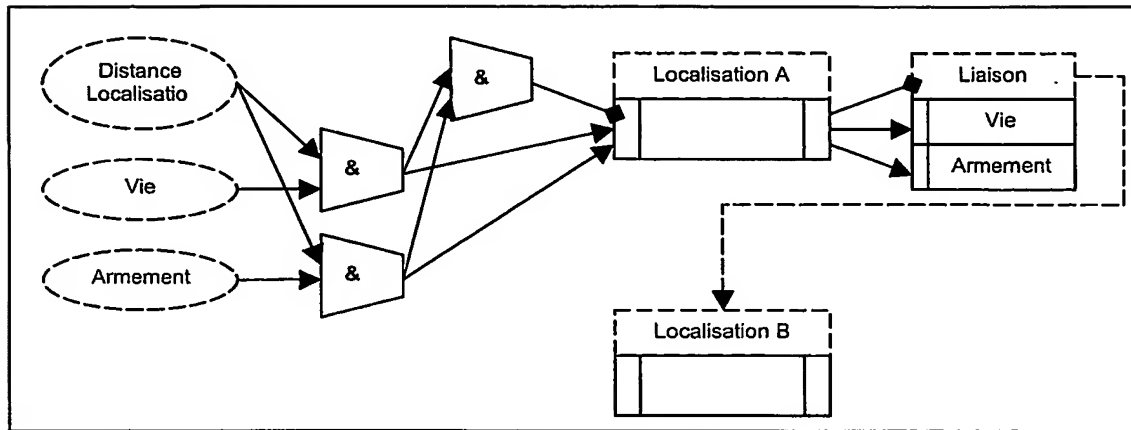


Figure 8

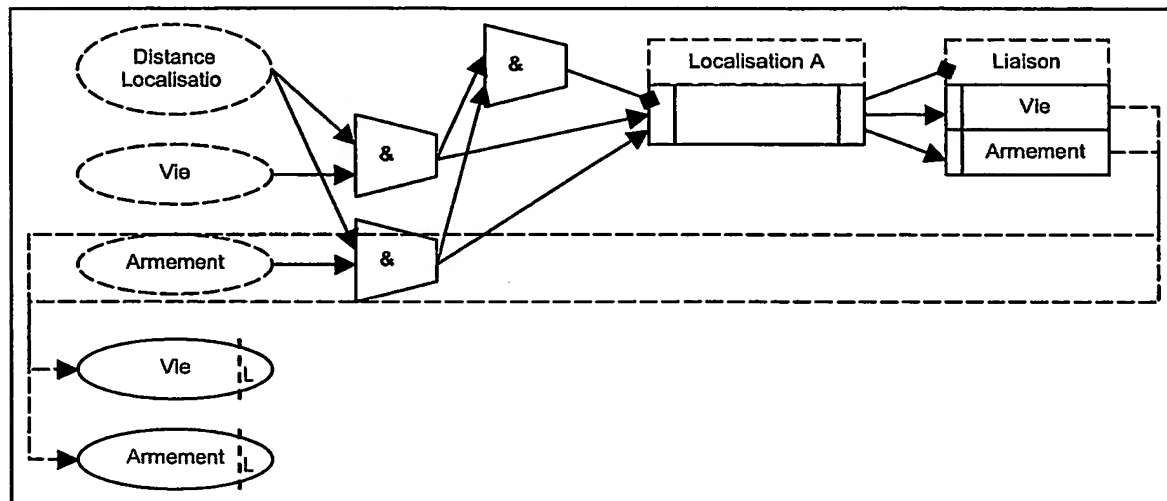


Figure 9